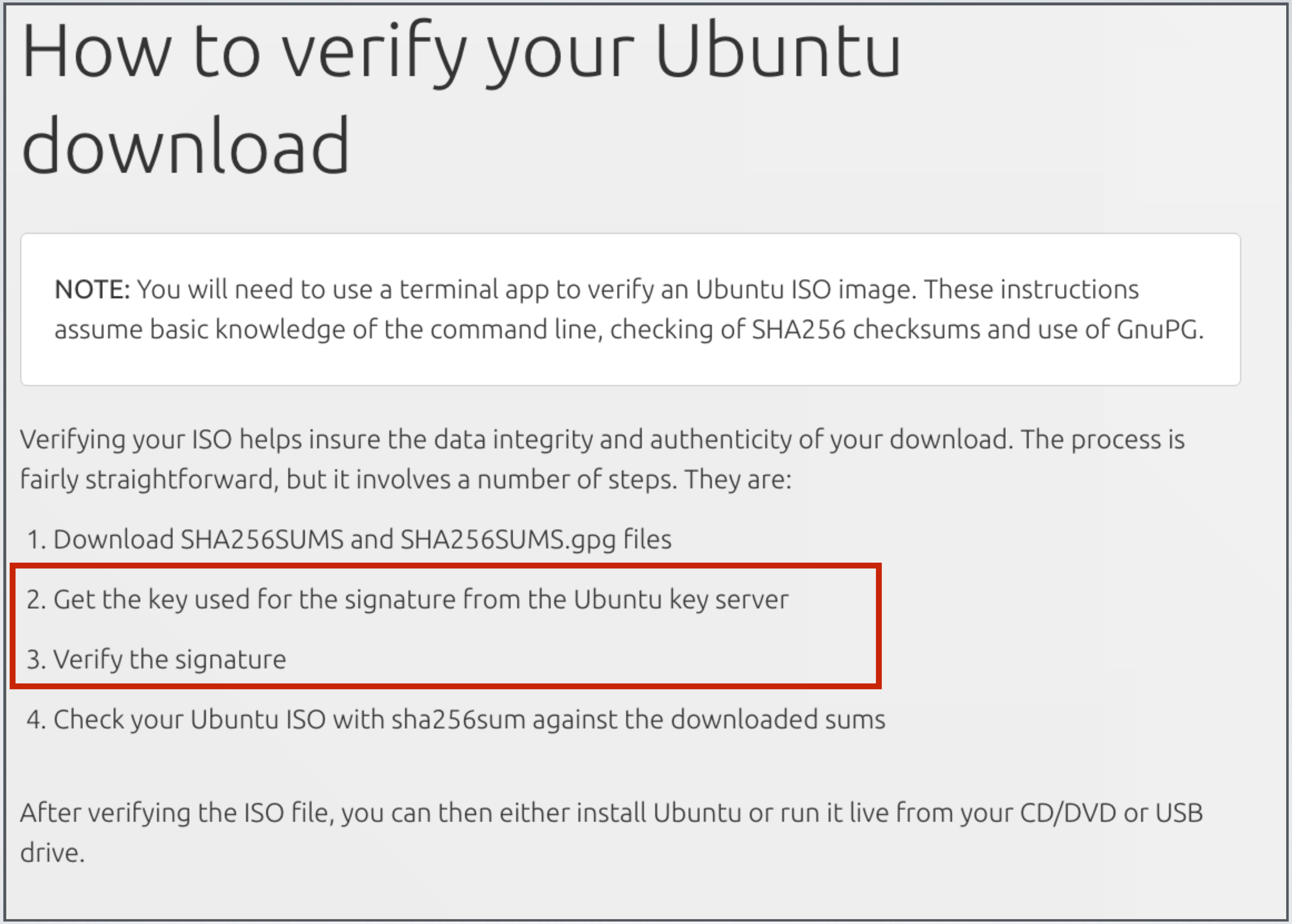
Digital Signatures

Thierry Sans



How to verify your Ubuntu download

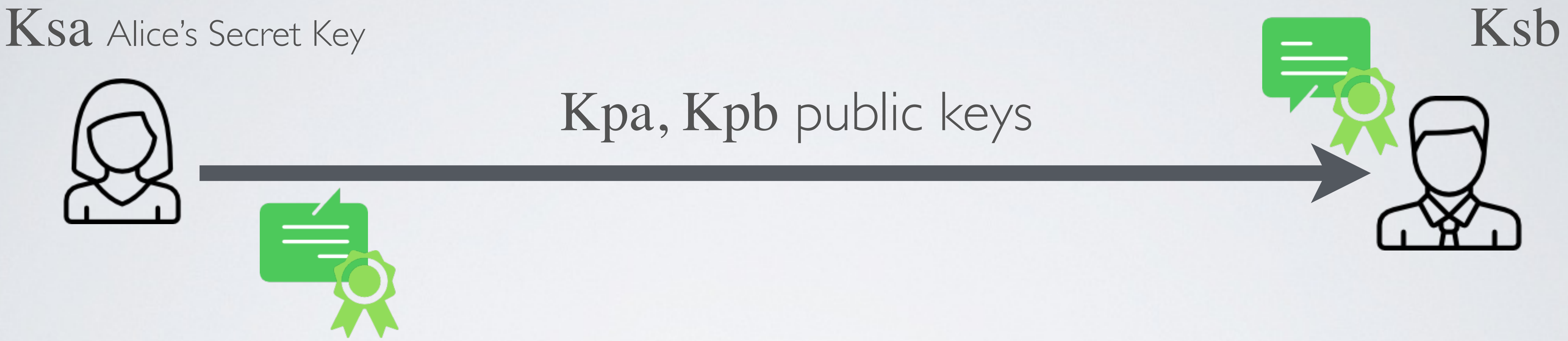
NOTE: You will need to use a terminal app to verify an Ubuntu ISO image. These instructions assume basic knowledge of the command line, checking of SHA256 checksums and use of GnuPG.

Verifying your ISO helps insure the data integrity and authenticity of your download. The process is fairly straightforward, but it involves a number of steps. They are:

1. Download SHA256SUMS and SHA256SUMS.gpg files
2. Get the key used for the signature from the Ubuntu key server
3. Verify the signature
4. Check your Ubuntu ISO with sha256sum against the downloaded sums

After verifying the ISO file, you can then either install Ubuntu or run it live from your CD/DVD or USB drive.

Digital Signature



➡ Use public cryptography to **sign and verify**

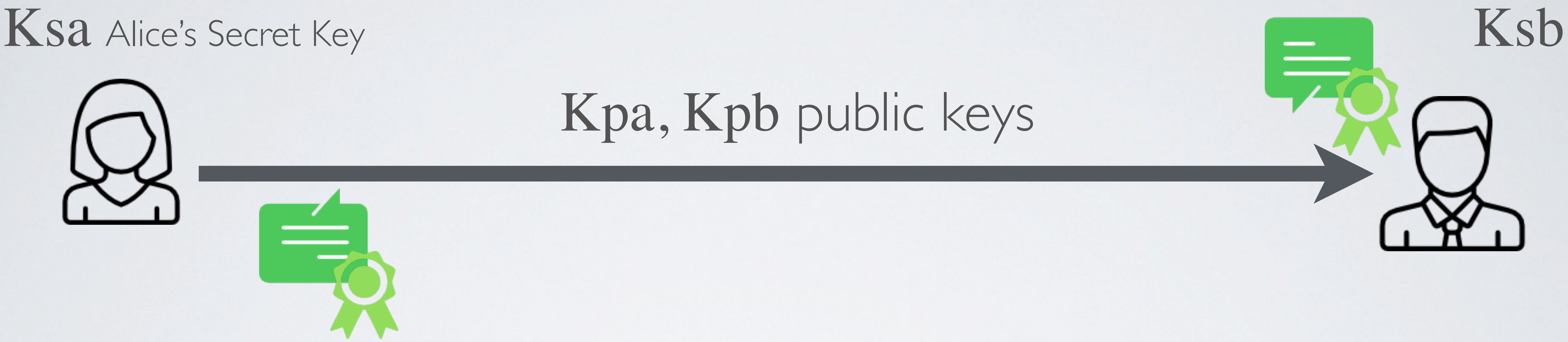
$$m \,||\, SIG_{Ksa}(m)$$

$$SIG_{Ksa}(m) = E_{Ksa}(H(m))$$

Non-repudiation as a special case of integrity

	MAC	Digital Signature
Integrity	✓	✓
Non-repudiation	✗	✓

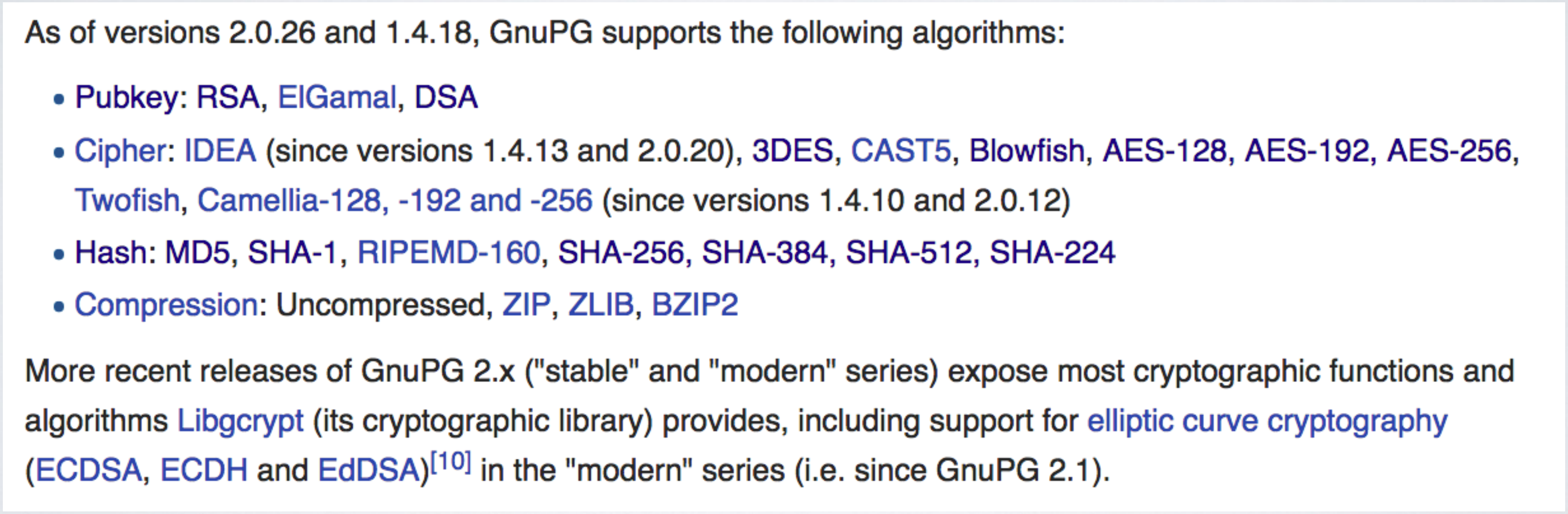
Digital Signatures and Confidentiality



1. Alice generates an asymmetric <u>session key</u> k
2. Use both symmetric and asymmetric cryptography to **encrypt, sign and verify** the message and the key

$$E_{Kpb}(k) \parallel E_k(m \parallel E_{Ksa}(H(m)))$$

This how GPG works

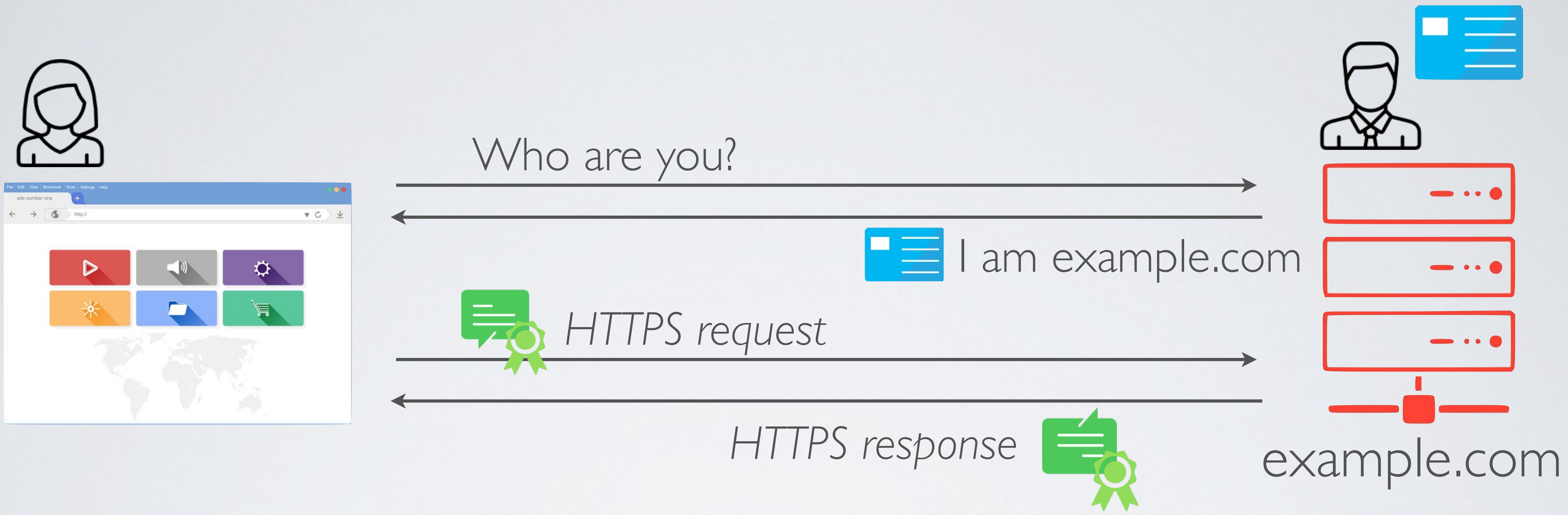

As of versions 2.0.26 and 1.4.18, GnuPG supports the following algorithms:

- Pubkey: RSA, ElGamal, DSA
- Cipher: IDEA (since versions 1.4.13 and 2.0.20), 3DES, CAST5, Blowfish, AES-128, AES-192, AES-256, Twofish, Camellia-128, -192 and -256 (since versions 1.4.10 and 2.0.12)
- Hash: MD5, SHA-1, RIPEMD-160, SHA-256, SHA-384, SHA-512, SHA-224
- Compression: Uncompressed, ZIP, ZLIB, BZIP2

More recent releases of GnuPG 2.x ("stable" and "modern" series) expose most cryptographic functions and algorithms Libgcrypt (its cryptographic library) provides, including support for elliptic curve cryptography (ECDSA, ECDH and EdDSA)[10] in the "modern" series (i.e. since GnuPG 2.1).

source *"GNU_Privacy_Guard"* on *Wikipedia*

This how HTTPS works



✓ **HTTPS = HTTP + TLS**

➡ TLS - Transport Layer Security (a.k.a SSL) provides

- **confidentiality :** end-to-end secure channel
- **integrity :** authentication handshake